

WS 2005/06

Mastermodul CS 5002

MSP: Methoden des Software-Entwicklungsprozesses

Teamentwicklung –
eXtreme Programming – Projekttagebuch

Prof. Dr. Klaus Quibeldey-Cirke
Fachhochschule Gießen-Friedberg

Forming – Storming – Norming - Performing

➤ Teamentwicklung nach Tuckman

1. Forming

- In der Orientierungsphase herrscht allgemein ein freundlicher Umgangston, die Gruppenmitglieder lernen sich kennen.
- Sie beginnen sich gegenseitig „abzutasten“ - Konflikte werden in der Regel vermieden.
- Die Anfangsbegeisterung ist noch groß.
- Die Leistungsfähigkeit ist mittelmäßig.



Bruce W. Tuckman:
[Developmental Sequence in Small Groups](#).
Psychological Bulletin, Bd. 63, Heft 6,
1965, S. 384-399.

Forming – Storming – Norming – Performing

2. Storming

- Bald darauf beginnt die Konfliktphase. Wettbewerb unter den Teammitgliedern setzt ein, verdeckte Konflikte treten zutage.
- Häufig werden lange Diskussionen über Vorgehensweisen und Methoden geführt.
- Jeder versucht, sein Terrain abzustecken.
- Die Leistungsfähigkeit sinkt rapide.



Forming – Storming – Norming – Performing

3. Norming

- Hat die Gruppe die Konfliktphase ausgetragen, kommt es in der Organisationsphase zur Teambildung – ein Wir-Gefühl entsteht.
- Teamregeln werden festgelegt (implizit oder explizit).
- Die Arbeitsleistung steigt an.



Forming – Storming – Norming - Performing

4. Performing

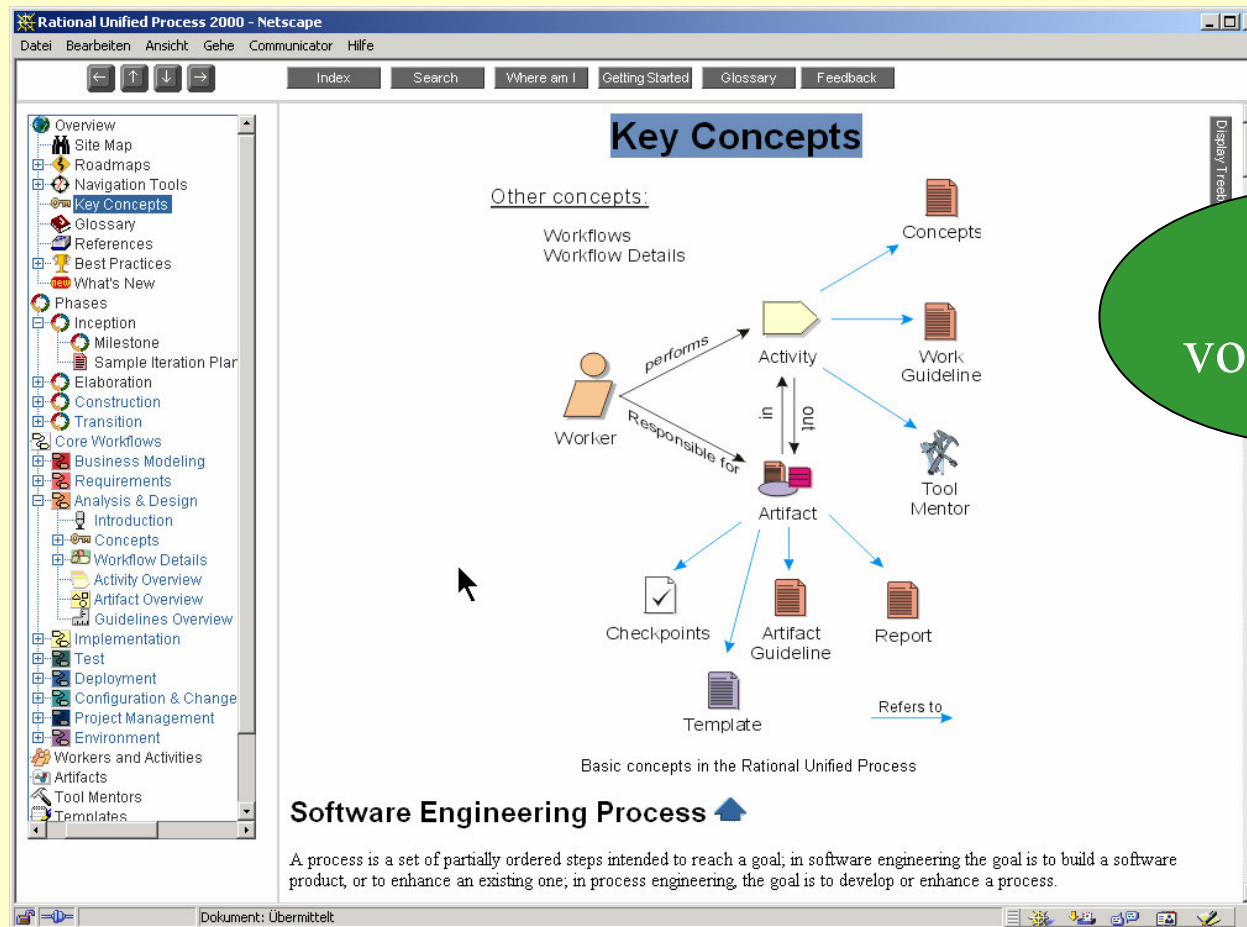
- Die Integrationsphase ist der eigentliche „Sollzustand“ einer Gruppe.
- Das Team kennt sich und vertraut sich.
- Das Team ist effizient und die Arbeitsleistung am besten.



Software-Vorgehensmodelle

Extrem I: Unified Process (UP)

- Schwergewicht: Unified Process in der [IBM-Rational-Ausprägung](#) (RUP)




Fokus
von SWT I

... Extrem II: eXtreme Programming (XP)

- Leichtgewicht: www.extremeprogramming.org

Extreme Programming: A Gentle Introduction.

The goal of this site is to provide an introduction and overview of Extreme Programming (XP). For a guided tour of XP follow the trail of little  buttons, starting here. Returning visitors can jump to [recent changes](#) to see what's new.

Let's begin with a simple question: [What is XP?](#) As you will see, it is a deliberate and disciplined approach to software development.

Next we might wonder [when to use XP](#). Risky projects with dynamic requirements are perfect for XP. These projects will experience greater success and developer productivity.

But do we need yet [another software methodology](#)? Actually we do. XP is a refreshing new approach. XP is successful because it emphasizes customer satisfaction and promotes team work.

So how could this possibly work? The most surprising aspect of XP is its [simple rules and practices](#). They seem awkward and perhaps even naive at first, but soon become a welcome change. Customers enjoy being partners in the software process and developers actively contribute regardless of experience level.

[What has changed here?](#) | [XP Practices and Rules](#) | [Email the webmaster](#)

Last modified April 15, 2001. See [recent changes](#).

Copyright (c) 1999, 2000, 2001 J. Donovan Wells. All Rights reserved.

Extreme Programming Project

The rules and practices must support each other. The [XP Map](#) shows how they work together to form a development methodology. Unproductive activities have been trimmed to reduce costs and frustration.

I want to try XP [how do I start?](#) Add a little to your current methodology or try it all at once. There is much here of benefit to any project.

Where can I [get more information](#)? There are conferences, books, web sites, and of course people. Many [people and projects](#) are looking to try XP want to join, need people?

What have other projects already [learned about XP](#)? Some important lessons learned.

[XP2001](#) conference on Extreme Programming in Italy, May 21-23.

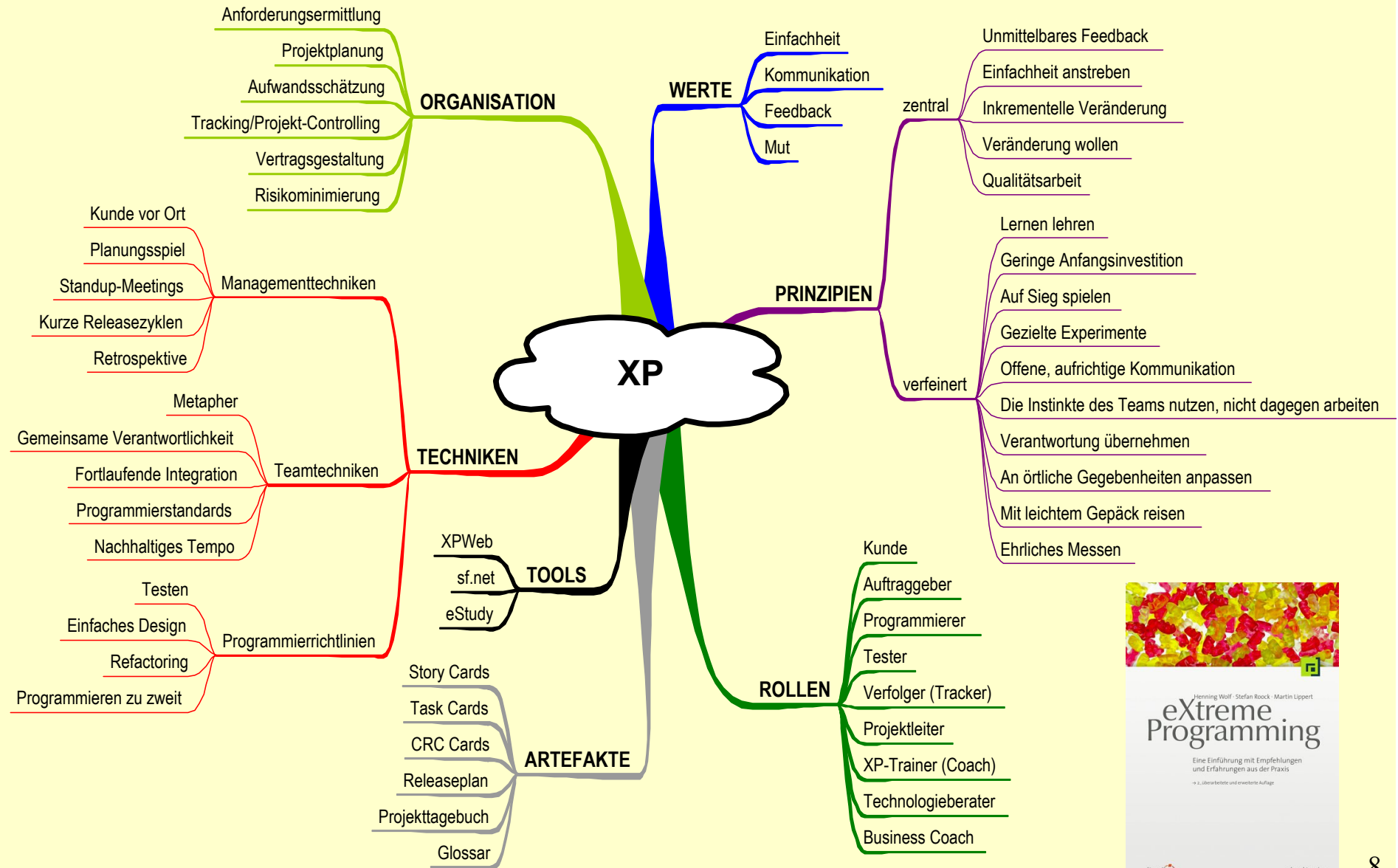
[XP Universe](#) July 23-25 in Raleigh, North Carolina.

Lessons Learned

Unser Fokus!

Die Themen der
folgenden Vorlesungen

Mind-Map zu eXtreme Programming



Projekttagbuch

- Artikel: [„Projektoptimierung mit Hilfe eines Projekttagbuchs“](#)
- Jedes Teammitglied ...
 - abonniert den eStudy-Forumsordner „PTB ...“
 - erstellt am Ende eines jeden Arbeitstages einen Beitrag:
 - Welche Aufgabe (Task) habe ich bearbeitet?
 - Welche Probleme traten auf?
 - Welche Probleme habe ich gelöst?
 - Welche Probleme konnte ich nicht lösen?
 - Welche Task werde ich am nächsten Arbeitstag bearbeiten?
 - aktualisiert am Ende eines jeden Arbeitstages in XPWeb:
 - die getätigten Aufwände seiner Tasks
 - die noch zu erbringenden Aufwände
 - redefiniert seine Tasks gegebenenfalls:
 - Jede Task sollte nicht mehr als einen Arbeitstag (= 4 Stunden) umfassen
- Der Teamleiter ...
 - ist verantwortlich für die Projektplanung in XPWeb und für das Führen des Projekttagbuchs:
 - geht von einem [Zeitbudget von 20 Arbeitstagen à 4 Stunden pro Mitglied](#) aus
 - plant und pflegt die User Stories in XPWeb
 - überwacht die Tasks und den Iterationsfortschritt in XPWeb
 - berichtet wöchentlich den Team-Status an den Projektleiter (Sascha Hemminger und KQC)

Prinzipielle Vorgehensweise der MSP-Teams

- Die MSP-Projektaufträge sind
 - alle gleich strukturiert:
 - Im Sinne der MSP-Lerninhalte geht es stets um die Konzeption und Realisierung eines Moduls **auf der komplexen Codebasis der Vorläuferprojekte**
 - Das bedeutet:
 - Sie müssen sich in die Struktur und Funktionalität der vorgefundenen Codebasis (CVS) selbstständig einarbeiten, und zwar soweit, wie es für Ihren Teamauftrag erforderlich ist. Das nennt man "**Reverse Engineering**".
 - Zu erstellende Artefakte: Use-Case-Modell, UML-Diagramme bzw. Programmablaufplan und Entwicklerdoku
 - Höchstwahrscheinlich finden Sie Codefragmente, die Sie zunächst überarbeiten müssen, bevor Sie neue Funktionen hinzufügen können. Das nennt man "**Refactoring**".
 - Besonderer Anspruch an die eStudy-Teams: **Security Audits**
 - Und das Ganze können Sie wegen der Komplexität der Projektaufträge und der Codebasis sowie des kappen Zeitbudgets nicht alleine innerhalb des Semesters leisten. Das geht nur per **Delegation und Teamarbeit**.

Ende der 2. Vorlesung

To Dos

1. Lektüre für alle ...

- Frank Westphal über eXtreme Programming
 - <http://www.frankwestphal.de/ExtremeProgramming.html>
- „Manifesto for Agile Software Development“
 - <http://www.agilemanifesto.org/>
- Scott W. Ambler über „Agile Unified Process“
 - <http://www.agilealliance.com/articles/>

2. Die Teamleiter ...

- initialisieren das Projekttagebuch
- pflegen die XPWeb-Rubriken
 - Team
 - Planung
 - Metapher und Links

3. Die Teams ...

- pflegen ihre User Stories und Tasks in XPWeb ein
- priorisieren und gewichten ihre User Stories beim nächsten Übungstermin mit dem Kunden
- eStudy-Teams verwenden
 - Mozilla Firefox (version 1.5)
<http://www.mozilla.org/projects/firefox/>
 - mit der Web Developer Extension (version 0.9.9) von Chris Pederick:
<http://chrispederick.com/work/webdeveloper/beta/>